



**AFRL-RY-WP-TR-2017-0071**

# **NETLIST-ORIENTED SENSITIVITY EVALUATION (NOSE)**

**Dr. Jeffrey Draper**

**University of Southern California  
Information Sciences Institute**

**MARCH 2017  
Final Report**

**Approved for public release; distribution is unlimited.**

*See additional restrictions described on inside pages*

**STINFO COPY**

**AIR FORCE RESEARCH LABORATORY  
SENSORS DIRECTORATE  
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320  
AIR FORCE MATERIEL COMMAND  
UNITED STATES AIR FORCE**

## NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nationals.

Copies may be obtained from the Defense Technical Information Center (DTIC)  
(<http://www.dtic.mil>).

AFRL-RY-WP-TR-2017-0071 HAS BEEN REVIEWED AND IS APPROVED FOR  
PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//Signature//

---

\*POMPEI L. ORLANDO, Program Manager  
Advanced Sensor Components Branch  
Aerospace Component & Subsystems Division

//Signature//

---

BRADLEY PAUL, Chief  
Advanced Sensor Components Branch  
Aerospace Component & Subsystems Division

//Signature//

---

TODD W. BEARD, Lt Col, USAF  
Deputy  
Aerospace Component & Subsystems Division  
Sensors Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

\*Disseminated copies will show “//Signature//” stamped or typed above the signature blocks.

<b>REPORT DOCUMENTATION PAGE</b>				<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>					
<b>1. REPORT DATE (DD-MM-YY)</b> March 2017		<b>2. REPORT TYPE</b> Final		<b>3. DATES COVERED (From - To)</b> 19 October 2015 – 19 October 2016	
<b>4. TITLE AND SUBTITLE</b> NETLIST-ORIENTED SENSITIVITY EVALUATION (NOSE)				<b>5a. CONTRACT NUMBER</b> FA8650-16-1-7601	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b> 61101E	
<b>6. AUTHOR(S)</b> Dr. Jeffrey Draper				<b>5d. PROJECT NUMBER</b> 1000	
				<b>5e. TASK NUMBER</b> N/A	
				<b>5f. WORK UNIT NUMBER</b> Y1E0	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  University of Southern California Information Sciences Institute 4676 Admiralty Way, Suite 1001 Marina del Rey, CA 90292				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  Air Force Research Laboratory Sensors Directorate Wright-Patterson Air Force Base, OH 45433-7320 Air Force Materiel Command United States Air Force				<b>10. SPONSORING/MONITORING AGENCY ACRONYM(S)</b> AFRL/Rydi	
				<b>11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)</b> AFRL-RY-WP-TR-2017-0071	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited.					
<b>13. SUPPLEMENTARY NOTES</b> This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This material is based on research sponsored by Air Force Research Laboratory (AFRL) and the Defense Advanced Research Agency (DARPA) under agreement number FA8650-16-1-7601. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory (AFRL) and the Defense Advanced Research Agency (DARPA) or the U.S. Government. Report contains color.					
<b>14. ABSTRACT</b>  The Netlist-Oriented Sensitivity Evaluation (NOSE) research effort is developing methodologies to assess sensitivities of alternative chip design netlist implementations. The research is somewhat foundational in that such measurement approaches are needed to provide an equitable basis for evaluating techniques aimed at enhancing robustness of designs. NOSE will highlight the circuit nodes in a design netlist that have the potential to cause the most significant impact on overall circuit behavior if any perturbation of any kind occurs at that node. Such information can lead to alternative implementations that are more resistant to such perturbations.					
<b>15. SUBJECT TERMS</b> chip design, complementary metal-oxide-semiconductor, netlist-oriented sensitivity evaluation					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT:</b> SAR	<b>18. NUMBER OF PAGES</b> 28	<b>19a. NAME OF RESPONSIBLE PERSON (Monitor)</b> Pompei L. Orlando <b>19b. TELEPHONE NUMBER (Include Area Code)</b> N/A
<b>a. REPORT</b> Unclassified	<b>b. ABSTRACT</b> Unclassified	<b>c. THIS PAGE</b> Unclassified			

## Table of Contents

Section	Page
List of Figures .....	ii
List of Tables .....	ii
1. PROJECT OVERVIEW .....	1
2. RESULTS .....	2
2.1 Netlist Sensitivity Results .....	2
2.2 Defect Analysis Results .....	4
3. SUMMARY/CONCLUSIONS .....	6
APPENDIX A: TOOLS TUTORIAL FOR GENERATING SCOAP VALUES .....	7
APPENDIX B: OUTBRIEF PRESENTATION .....	8
LIST OF ACRONYMS, ABBREVIATIONS, AND SYMBOLS .....	23

## List of Figures

Figure	Page
Figure 1: Defects Resulting in an Electrically and Logically Stable CMOS Function .....	4
Figure 2: Common Implementation of 2-Input XOR Function .....	5

## List of Tables

Table	Page
Table 1. Example of Design Trade-Offs involving Controllability and Observability for Alternative Implementations of a 4-bit Multiplier.....	3

## **1. PROJECT OVERVIEW**

The primary objective of the Netlist-Oriented Sensitivity Evaluation (NOSE) project was to develop methodologies to assess sensitivities of alternative chip design netlist implementations. The conducted research is somewhat foundational in that such measurement approaches are needed to provide an objective basis for evaluating techniques aimed at enhancing robustness of designs.

The design effort leveraged prior work in design-for-testability and single-event transient analysis to devise a methodology for scoring the sensitivity of circuit nodes in a netlist and thus providing the raw data for any meaningful statistical analysis such as histograms. One challenging part of the research plan was to determine the best metric for measuring sensitivity that would also point to alternative netlist implementations with differing sensitivities. As detailed below, existing work in the design for testability realm greatly aided in providing some metrics.

NOSE therefore exhibited the following innovative claims:

- Metrics that capture the sensitivity of circuit nodes in a chip design netlist implementation
- Methodologies for quantifiably measuring the sensitivity of chip design netlist implementations
- Heuristics for generating an alternative netlist implementation for the same logic function to achieve a better robustness measure

## 2. RESULTS

After discussion in late March 2016, the NOSE project pursued two paths for the netlist sensitivity study:

1. Tools to calculate sensitivity of netlist nodes (signals) and heuristics for generating alternative netlist implementations yielding differing sensitivity metrics.
2. Defining the acceptable effects of defects and the minimum number of defects needed to cause an acceptable effect.

### 2.1 Netlist Sensitivity Results

For the calculation of signal sensitivities, we pursued the specification and/or development of tools to derive the 1-controllability and 0-controllability and observability for every node (signal) in a combinatorial netlist. We had earlier suggested that the measure of 1-controllability for a particular signal be defined as the minimum number of primary input values that have to be set to a known value (1 or 0) to cause the signal to have a logic value of 1, similarly for 0-controllability for the signal to have a logic value of 0. We had also suggested that the measure of observability for a particular signal is the minimum number of primary input values that have to be set to a known value to enable that signal's state to be observable at a primary output.

Over the course of the NOSE project, we investigated existing tools to see how especially design-for-testability (DFT) tools could be applied to this problem. From a DFT point of view, controllability is measured by whether both '0' and '1' are able to propagate to each and every node within the target patterns. A point is said to be controllable if both '0' and '1' can be propagated through scan patterns. Most currently available DFT tools employ various levelling algorithms to find the controllability and observability of nets. TetraMAX from Synopsys is an example which displays controllability and observability for all nodes. These tools use the SCOAP measures to rate the nodes, as defined in "SCOAP: Sandia Controllability/Observability Analysis Program", Lawrence Goldstein and Evelyn Thigpen. Controllability values are ranked from 1 to infinity, with 1 being the easiest to control, from the perspective of primary inputs. Observability values are ranked from 0 to infinity, with 0 being the easiest to observe, from the perspective of primary outputs. The logic cone is evaluated from all inputs and the controllability and observability values are computed using Goldstein's SCOAP measures for individual gates. If a node cannot be observed or controlled, additional control points are added which can be included in the scan chain for testing.

Therefore, we can use these existing tools to provide SCOAP measures for nodes between alternative implementations of the same logic to determine which implementation provides the most desirable SCOAP measure for both controllability and observability. This approach was tested on two implementations of a 4 bit multiplier: one having a Ripple Carry adder (RCA) as a final stage and the other having a Kogge Stone adder (KSA) as the final stage. Analysis from TetraMAX reveals the former to have C1-C0-O to be 4-4-4 while the latter has 4-6-6, where C0 = min inputs needed to control the pin to a 0, C1 = control the pin to a 1, O = observe the value at the pin.

A summary of the results is shown in Table 1.

**Table 1. Example of Design Trade-Offs involving Controllability and Observability for Alternative Implementations of a 4-bit Multiplier**

Implementation	Area ( $\mu\text{m}^2$ )	Energy (nJ)	Gates	1 Ctrlability	0 Ctrlability	Observability
RCA stage	93.9	0.3	37	4	4	4
KSA stage	113.6	0.4	49	4	6	6

We can observe from the results that adding more complexity has a direct impact on the controllability and observability of the nodes in the design. Although the KSA has only 6 levels of logic compared to the RCA, it has more complex gates compared to the primitive ones of the RCA, resulting in higher controllability and observability values. Therefore we see a tradeoff between speed and controllability right away. Appendix A provides a brief tools tutorial showing how to obtain these values using TetraMAX.

Another observation while analyzing netlists of varying sizes was that as the designs get more complex, the Automatic Test Pattern Generation of TetraMAX cannot attain full coverage for testing. In those cases, the SCOAP values are not populated in the netlist. This can possibly be circumvented by having just a dedicated SCOAP calculating program, which we haven't managed to obtain but should be implementable.

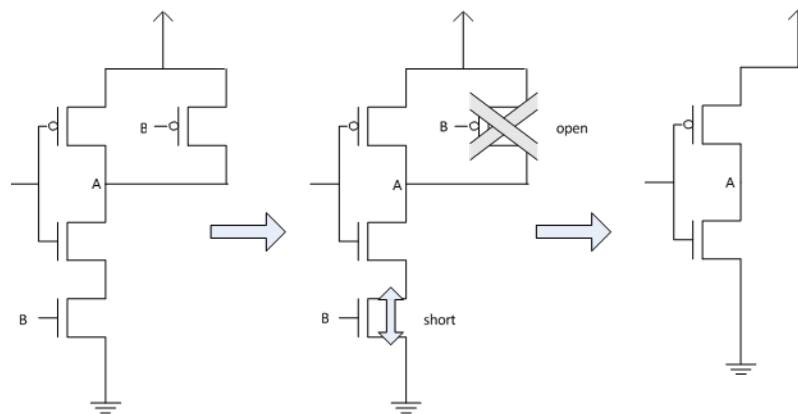
Additionally, a secondary approach to determine which set of inputs effect the result of a node is to back-compute the complete expression tracing to the inputs. In other words, one could evaluate the complete logic cone starting from the node of interest, and moving towards the inputs of the combinational block. Once the complete logic cone to the input is evaluated, Boolean reduction methods like the Karnaugh Map or Quine-McCluskey algorithm can be used to minimize the expression to prime implicants. The number of distinct variables in the minimized expression is the number of inputs that need to be set for (1 or 0) controllability. The exact number of inputs needed for 1 and 0 controllability will depend on the exact reduced expression as well. For example if the final expression is  $(a \vee (b \vee c) \wedge (d \vee e \vee f))$ , the 1-controllability is just one ( by setting  $a=1$ ). However, the 0-controllability is three ( $a, b$  and  $c$  must be set to 0). While we found no tools which currently provide this computation, parsing netlists to evaluate logic cones can be done easily with tools and scripts.

Some work was done earlier in this general direction for determining sensitivities for radiation hardening techniques. We had previously developed a perl script (using veriperl) to parse a verilog netlist and make a graph which could be processed by faster high level languages. The graph can be read by other tools, and various analytics can be performed. Previously all possible paths from the inputs to the outputs were traced, and the most critical node in the netlist was determined based on parameters like gates associated with the node, location on the chip, fanout etc. Such a parser could be used as the front-end for the logic evaluation described above if more detailed info beyond the SCOAP measures from DFT tools is needed.



## 2.2 Defect Analysis Results

For analyzing the effects of defects, we confine the acceptable defect space to those which result in electrically and logically correct complementary metal-oxide-semiconductor (CMOS) functions. With this constraint as a requirement, we began our analysis by first considering simple n-input functions (NAND, NOR), such as the 2-input NAND gate shown in Figure 1. Given the dual nature of pull-up and pull-down functions in simple gates (e.g., a parallel connection of pfets corresponds to a series connection of nfets for corresponding inputs and vice versa), whatever effects defects cause in one section (pull-down or pull-up) of the gate, other defects must be present in the opposite section to maintain the dual nature of the pull-up and pull-down functions. For simple gates, a minimum of two fets must be altered to maintain an acceptable CMOS function. The generalization is that for any fet in a series connection where defects cause the fet to be permanently ON, the fet in the opposite section (pull-down or pull-up) which corresponds to the same input and must be in a parallel network due to the duality of CMOS, must be permanently OFF. With this approach, it can be shown that such defects can effectively result in the elimination of an input from any logic function. For example, an inverter can be converted to a pull-up or pull-down function, depending on which fet is shorted and which fet is opened. N-input nand/and/nor/or gates can be converted to (N-1)-input nand/and/nor/or, respectively. This relationship between the number of defects and the number of inputs removed from the function can also be generalized; the reduction of x inputs (up to N-1) from a logic cell requires defects that affect 2x fets in the logically consistent manner described above.

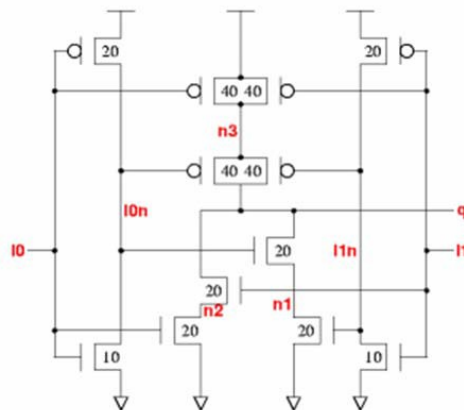


**Figure 1: Defects Resulting in an Electrically and Logically Stable CMOS Function**  
(Example: 2-input NAND transformed into an inverter)

Given this general logical principle, we now analyze what defects could result in the effect of a fet being permanently OFF or ON. For the ON case, the drain could be shorted to source or the gate could be pulled high(low) for a nfet(pfet). In the former case, the short of drain to source should be done in a manner that minimizes any adverse electrical effects. In the latter case, two defects would be required to maintain acceptable electrical properties: an open in the original gate connection and short of the gate to the proper Vdd/GND rail. Similarly for the OFF case, two defects are needed to open the original gate connection and then short the gate to the appropriate Vdd/GND rail that maintains the fet in a permanently OFF state. It could also be possible that a defect could destroy the channel to create an open, but doing so while still

preserving the gate characteristics so that the gate-driving signal is not affected could be challenging. It should be noted that in both cases for defects on the gate creating effects of fet opens or shorts, the open of the original gate connection will present as a slightly lower capacitive load to the original driver, but this is unlikely to result in significantly differently electrical properties. It is also unlikely to result in hold timing violations, but it is a possibility for defects occurring on extremely short paths between clocked elements and/or if sufficient margins do not exist in the original design.

While analysis of simple gates is rather straightforward, as shown above, compound gates, such as XOR, AOI, and OAI, are a class that requires special treatment. As it turns out, many standard cell library implementations for such compound gates still follow a fairly strict dual structure. For example, a common implementation for a 2-input XOR gate is shown below in Figure 2. As can be seen, the core structure of the gate beyond the inversion of the two primary inputs contains a compound structure where the dual pull-up and pull-down sections are implemented in a very straightforward manner where parallel connections in one section correspond to series connections in the other section, and vice versa. Hence, the methodology described earlier for simple logic gates can still be applied to this type of gate to obtain alternative functions. For example, if the pfet driven by  $i0n$  were “opened” and the nfet driven by  $i0n$  were shorted, a stable CMOS function simplified to (not  $i0$  and  $i1$ ) would result. In fact, such compound gates with inherent dual structures provide a rich palette of possible alterations with non-standard resulting functions.



**Figure 2: Common Implementation of 2-Input XOR Function**  
(<http://www.vlsitechnology.org/html/cells/sxlib013/xr2.html>)

So clearly, the methodology described earlier can be generalized to any implementation where dual structures in the pull-down and pull-up sections of a CMOS gate separate into series and parallel connections from the perspective of any particular input.

### **3. SUMMARY/CONCLUSIONS**

The NOSE project pursued two paths for the netlist sensitivity study:

1. Tools to calculate sensitivity of netlist nodes (signals) and heuristics for generating alternative netlist implementations yielding differing sensitivity metrics.
2. Defining the acceptable effects of defects and the minimum number of defects needed to cause an acceptable effect.

While this study developed the groundwork for both paths, there is clearly more that could be done in follow-on research. For the netlist sensitivity research, more formal approaches for heuristics to generate alternative netlist implementations with predictable resulting controllability and observability values would be helpful. For instance, using SCOAP values as metrics, the bounds appear to indicate that the most levels of logic for implementing a function would provide the highest control- lability values. One approach for attempting to accomplish this would be to restrict synthesis to using 2-input gates only, but there is no guarantee that the resulting netlist would have more levels of logic than alternative implementations. For example, it could be that the 2-input gates are arranged in a logarithmic logic cone while alternatives using more complex gates could result in more of a series cone. More analysis is needed in this area.

Similarly, for the defect analysis, more work is needed for esoteric complex gates. The approach shown above is obvious for classical CMOS gates where pull-down and pull-up structures are straightforward duals (fets in series in pull-down correspond to fets in parallel in pull-up and vice versa). Some very complex library cells are not always implemented so cleanly, and additional analysis is needed to study, characterize, and formulate an approach for such cells.

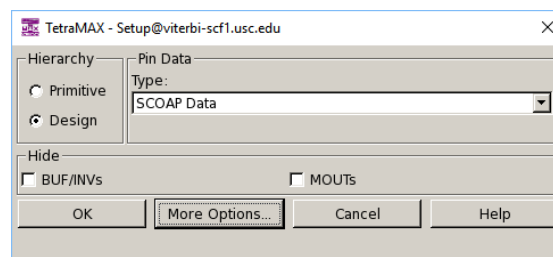
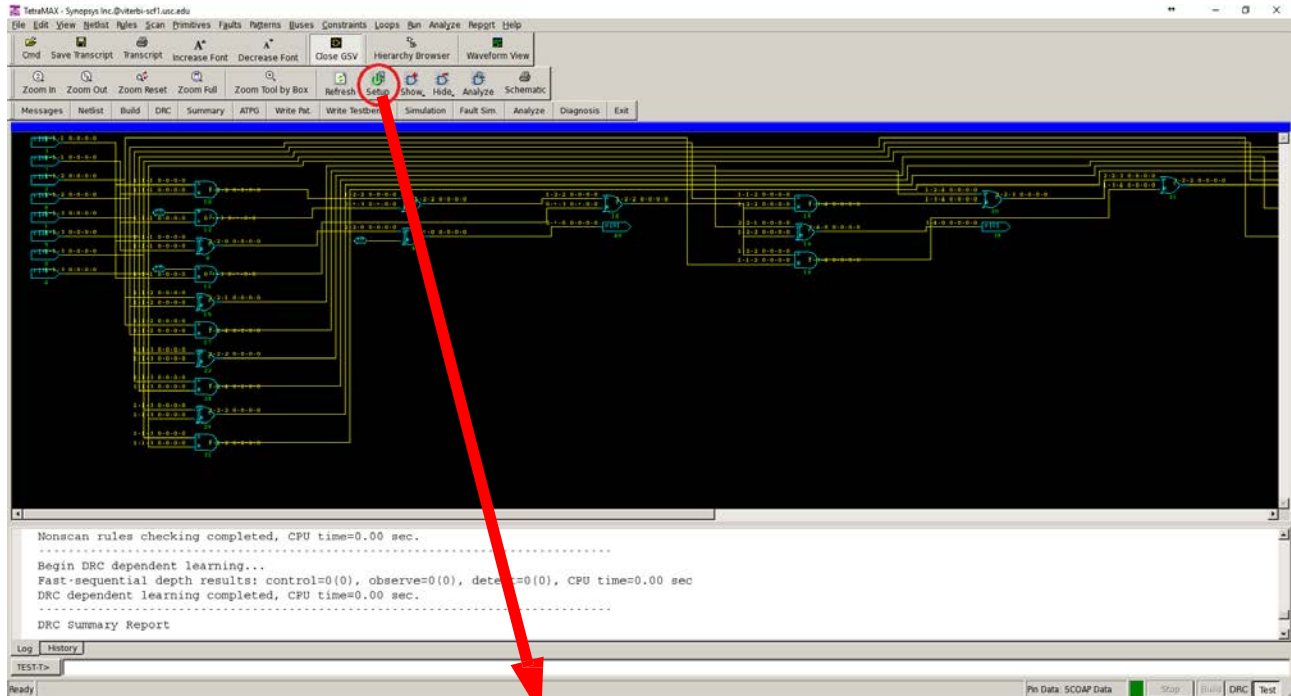
Appendix B provides the outbrief slides presented to Kerry Bernstein November 4, 2016 at the Defense Advanced Research Projects Agency (DARPA).

## APPENDIX A: TOOLS TUTORIAL FOR GENERATING SCOAP VALUES





TetraMAX Script to run scoap.




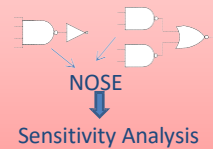
```
read_netlist /NCSU45PDK/FreePDK45/osu_soc/lib/files/gscl45nm.v -library  
read_netlist /netlist/multiplier/netlist/UBRCA_3_0_3_0.syn.v  
run_build_model UBRCA_3_0_3_0  
run_drc
```

TetraMAX GUI setup for SCOAP display



## APPENDIX B: OUTBRIEF PRESENTATION

	
<h3><i>Netlist-Oriented Sensitivity Evaluation (NOSE)</i></h3> <h4><i>November 4, 2016 Update</i></h4>	
<p><b>Jeff Draper</b>  <b>Deputy Director of Computational Sciences and Technology, ISI</b>  <b>Research Associate Professor, Ming Hsieh Department of EE</b>  <a href="mailto:draper@isi.edu">draper@isi.edu</a></p>	
 <p>4676 Admiralty Way Marina del Rey, CA</p>	 <p>3811 N Fairfax Drive Arlington, VA</p>
<p>11/04/2016</p>	
<div>   </div>	

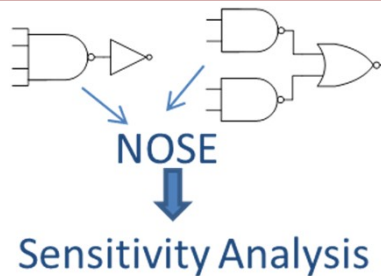
 <h3>Netlist-Oriented Sensitivity Evaluation (NOSE)</h3> 	
<p style="text-align: center;"><b>Motivation</b></p> <ul style="list-style-type: none"> <li>Despite a long history of numerous efforts to address hardware validation, there are still areas where better analysis tools would provide insight.</li> <li>To better quantify the effect of any proposed technique, foundational research is needed that: <ul style="list-style-type: none"> <li><b>Characterizes chip design netlists to identify the circuit nodes that have the greatest impact on circuit sensitivity</b></li> <li><b>Develops a methodology for generating alternative design netlist implementations to achieve a stated sensitivity goal</b></li> </ul> </li> </ul>	<p style="text-align: center;"><b>Team</b></p> <p><u>University of Southern California</u>  <u>Information Sciences Institute</u>  PI: Jeff Draper  <a href="mailto:draper@isi.edu">draper@isi.edu</a>  (310) 448-8750</p>  
<p style="text-align: center;"><b>Project Goals</b></p> <ul style="list-style-type: none"> <li>Develop models to quantify overall sensitivity measures of a particular chip netlist implementation and provide a sensitivity histogram for nodes in the netlist <ul style="list-style-type: none"> <li>Models are expected to leverage controllability and observability characteristics of design-for-testability approaches</li> </ul> </li> <li>Using DeMorgan's theorem or other similar Boolean Algebra techniques, demonstrate how alternative netlist implementations can alter sensitivity</li> </ul>	
<p><b>Research Plan</b></p>	
<p style="text-align: center;"><b>Task Plan</b></p> <ul style="list-style-type: none"> <li>Task 1: Preliminary definition of metrics for quantifying netlist sensitivity (2 Months)</li> <li>Task 2: Development of NOSE methodology for measuring sensitivity (6 months)</li> <li>Task 3: Demonstration and evaluation of NOSE on sample design (4 months)</li> <li>Overall project duration: 12 months</li> <li>Overall project budget: ~\$300K</li> </ul>	<p style="text-align: center;"><b>Metrics</b></p> <ul style="list-style-type: none"> <li>Sensitivity of netlist alternatives versus respective measures of area, speed, energy</li> <li>Runtime of tools used to quantify sensitivity</li> <li>Measure of the number of circuit nodes in a netlist exceeding a specified sensitivity threshold</li> </ul>

## NOSE Primary Objectives



- Metrics that capture the sensitivity of circuit nodes in a chip design netlist implementation
- Methodologies for quantifiably measuring the sensitivity of chip design netlist implementations
- Heuristics for generating an alternative netlist implementation for the same logic function to achieve a better robustness measure

## NOSE Example



- Both circuits implement a 4-input AND function
- The only netlist differences are the types of gates, the number of gates, and the number of internal signals
- Do these characteristics point to a metric(s) that indicate something about sensitivity?

## NOSE Additional Objectives



- **Define acceptable effects of defects and the minimum number of defects needed to cause an acceptable effect.**
  - Confine the acceptable defect space to those which result in electrically and logically correct CMOS functions.
- **Use common cells in standard cell libraries for analysis.**

## NOSE PRIMARY OBJECTIVES STATUS



## Netlist-Oriented Sensitivity Background



- Best sensitivity metrics at the netlist level involve concepts of controllability and observability
- Rich history of these concepts as involved in testability can be leveraged for our purposes
- One particularly relevant methodology involves SCOAP metrics
  - SCOAP (Sandia Controllability/Observability Analysis Program)
    - Goldstein, L. H. (1979). Controllability/observability analysis of digital circuits, IEEE Transactions on Circuits and Systems, 26, 685-693.
    - Lawrence Goldstein and Evelyn Thigpen, SCOAP: Sandia Controllability/Observability Analysis Program, DAC 1980

## Measuring SCOAP using Synopsys TetraMax





## Outline



- Background
- SCOAP
- Altering Controllability and Testability
- Using TetraMax for Validation
- Example

## Background



- **Controllability is defined as the difficulty of setting a particular logic signal to a 0 or a 1.**
  - Primary inputs(PI) are free (usually assigned a value of 1).
- **Observability defined as the difficulty of observing the state of a logic signal.**

### Purpose:

- **Analysis of difficulty of testing internal circuit nodes.**
  - May need to modify circuit, add observation points or test hardware.
- **Can be used to guide ATPG algorithms, i.e., to help them make decisions by providing information about the difficulty of setting signals.**
- **Can be used to estimate fault coverage.**
- **Can be used to estimate test vector length.**

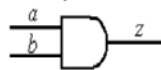
## SCOAP



- **Controllability Metrics: The basic process: Set PIs to 1, progress from PIs to POs, add 1 to account for logic depth.**



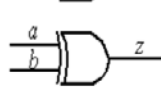
$$\begin{aligned} CCO(z) &= CCI(a) + 1 \\ CCI(z) &= CCO(a) + 1 \end{aligned}$$



$$\begin{aligned} CCO(z) &= \min\{CCO(a) + CCO(b)\} + 1 \\ CCI(z) &= CCI(a) + CCI(b) + 1 \end{aligned}$$



$$\begin{aligned} CCO(z) &= CCO(a) + CCO(b) + 1 \\ CCI(z) &= \min\{CCI(a) + CCI(b)\} + 1 \end{aligned}$$



$$\begin{aligned} CCO(z) &= \min\{CCO(a) + CCO(b), CCI(a) + CCI(b)\} + 1 \\ CCI(z) &= \min\{CCI(a) + CCO(b), CCO(a) + CCI(b)\} + 1 \end{aligned}$$

Information Sciences Institute

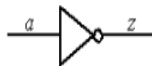
11

USC Viterbi  
School of Engineering

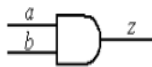
## SCOAP



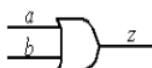
- **Observability Metric: The basic process: After controllabilities computed, set POs to 0, progress from PO to PIs, add 1 to account for logic depth.**



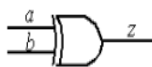
$$CO(a) = CO(z) + 1$$



$$CO(a) = CO(z) + CCI(b) + 1$$



$$CO(a) = CO(z) + CCO(b) + 1$$



$$CO(a) = CO(z) + \min\{CCO(b) + CCI(b)\} + 1$$

Information Sciences Institute

12

USC Viterbi  
School of Engineering

## Altering Controllability and Testability



- **Type of gate and logic depth affects SCOAP values**
  - Different synthesis runs with different optimizations will lead to different SCOAP values for the same node

## TetraMAX



- **After test pattern generation, SCOAP values of nodes can be labelled in TetraMAX.**
- **Values are generated during the preprocess section.**
- **Two ways to display data**
  - FULL\_SEQ\_Scoap\_data - the pin data field shows the set of SCOAP controllability and observability numbers for the pin
  - SCOAP Data - the displayed data is the SCOAP rating value. There are two sets of numbers, the first set consists of three characters of the form "C0-C1-O" and is for combinational ATPG. The the second set consists of four characters of the form "C0-C1-O-D" and is for sequential ATPG.

## TetraMAX



When the pin data setting of SCOAP is selected, the displayed data is the SCOAP rating value. There are two sets of numbers, the first set consists of three characters of the form "C0-C1-O" and is for combinational ATPG. The second set consists of four characters of the form "C0-C1-O-D" and is for sequential ATPG.

For the "C0-C1-O" format, each field is the minimum number of scan cells or input ports needed to:

C0 = control the pin to a 0.

C1 = control the pin to a 1.

O = observe the value at the pin.

For the "C0-C1-O-D" format, each field is the minimum sequential depth necessary to:

C0 = control the pin to a 0.

C1 = control the pin to a 1.

O = observe the value at the pin.

D = sensitize the gate to detect the fault at an observe point.

An asterisk "\*" indicates the value exceeds the 254 number program limit for tracking this information.

*Information Sciences Institute*

15

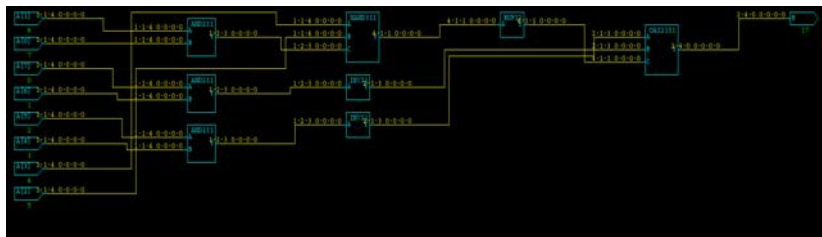
**USC Viterbi**  
School of Engineering

## Example Baseline



- To test change in SCOAP value for simple logic with a bunch of logic gates.

The 1<sup>st</sup> run was made with all the Design compiler optimizations ( command compile\_ultra)



Since it is combinational logic, only the 1<sup>st</sup> 3 SCOAP metrics are populated.

*Information Sciences Institute*

16

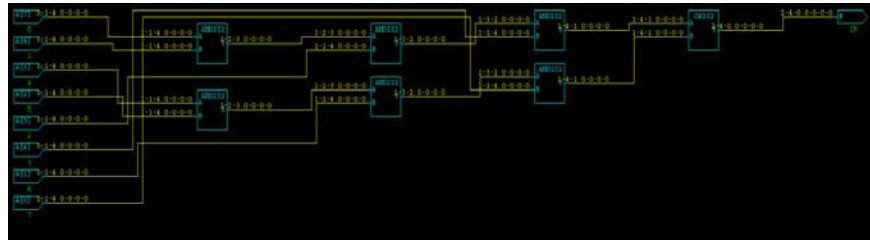
**USC Viterbi**  
School of Engineering

## Example Resynthesis



- To test change in SCOAP values for simple logic with a bunch of logic gates.

The 2<sup>nd</sup> run was made with strict design only synthesis ( command compile -only\_design rule)



This forces logic gate usage confirming to the Verilog input file, leading to different SCOAP values

## Example Summary

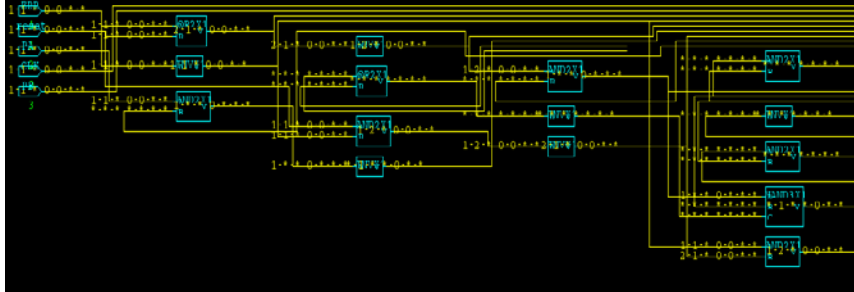


- Based on how the preceding logic is implemented the range of values for the whole design can be different, as well as SCOAP values of any individual node .
- Overall design comparison
  - Max values for optimized gates version
    - C0 - 4 ( input of final OR)
    - C1 - 4 ( output )
    - O - 4 ( input)
  - Max values for primitive gates version
    - C0 - 2 (output)
    - C1 - 4 (output)
    - O - 4 (input)
- Individual node example
  - SCOAP values of input of the final gate change from 2-1-3 to 1-4-1 going from optimized to primitive implementation

## Large Designs



- When testing large designs the preprocessor fails to populate the values.



- This has a strong correlation with the test pattern coverage that TetraMAX generates when completing ATPG.

## Future Work



- **Need to develop an approach for circumventing challenges related to the SCOAP metrics calculation being intertwined with TetraMax**
  - Current set of tools unwieldy on large circuits because full coverage is needed for calculation of SCOAP values
  - Tool probably already exists if we can get to the right people, e.g., the original SCOAP program
  - Alternatively, need more experimentation with TetraMax options to drive convergence
- **Need to better formalize recipe for affecting SCOAP values**
  - We have shown how different implementations affect SCOAP values but ideally would want to quantify predictable results based on implementation methods
    - For example, will using 2-input gates exclusively always increase the max SCOAP values for a logic block?



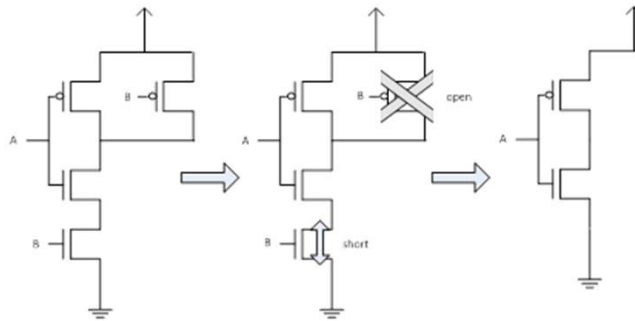
## NOSE DEFECT EFFECT ANALYSIS



## Defect Effect Analysis Background

- **Start with considering n-input simple functions (NAND, NOR)**
  - Can regard inverters as simple 1-input functions
- **Given the dual nature of pull-down and pull-up networks in simple CMOS logic functions, any defects that result in a different CMOS logic function must affect both the pull-up and pull-down networks in a logically consistent manner**
  - Shorting of a series fet in either side (pull-up or pull-down network) of the gate must be matched by opening of the corresponding fet on the other side of the gate (which must be in a parallel network given the dual nature of CMOS logic)

## Example: 2-input NAND



- **If defects occur in a manner to short an nfet and open the corresponding pfet, an inverter results**
  - Minimum of 2 fets must be affected in this manner to result in a logically consistent CMOS function

## Generalizations



- **Can generalize to n-input simple gates; with defects occurring in this manner, any n-input simple gate will effectively behave as an (n-1)-input gate**
- **Can also generalize the relationship between number of defects and number of input reduction**
  - Reduction of x inputs (up to n-1) requires defects that affect 2x fets in the logically consistent manner described



## Impact of Electrical Considerations



- For each fet to be shorted or opened in a manner with minimal impact to the electrical behavior of the circuit results in further constraints
- For a fet to be shorted, the original wire connected to the gate must be opened and the gate connection must then be shorted to Vdd (Vss) for nfets (pfets)
  - Similarly for a fet to be opened, the original wire connected to the gate must be opened and the gate connection must then be shorted to Vss (Vdd) for nfets (pfets)

## Timing Considerations



- Severing the original wire to a fet gate will present a lighter capacitive load to the original driving signal
  - Speed-up of original gate-driving signal likely to be insignificant in most cases (i.e., a single gate load is an insignificant contributor to the overall capacitive load of most nets in a state-of-the-art design)
  - However, for a single-loaded signal on a short wire, the effect could be significant
    - Only a concern if the affected signal is on a very short path between clocked elements
      - Could cause a hold violation



## Future Work



- **Approach is obvious for classical CMOS gates where pull-down and pull-up structures are straightforward duals (fets in series in pull-down correspond to fets in parallel in pull-up and vice versa)**
- **Some very complex library cells are not always implemented so cleanly**
  - Need to study, characterize, and formulate an approach for such cells

## **LIST OF ACRONYMS, ABBREVIATIONS, AND SYMBOLS**

<b>ACRONYM</b>	<b>DESCRIPTION</b>
AFRL	Air Force Research Laboratory
CMOS	complementary metal-oxide-semiconductor
DARPA	Defense Advanced Research Projects Agency
DFT	design-for-testability
KSA	Kogge Stone adder
NOSE	Netlist-Oriented Sensitivity Evaluation
RCA	Ripple Carry adder
SCOAP	Sandia Controllability/Observability Analysis Program